Amendments to the Claims

The following listing of the claims will replace all prior versions, and listings of the claims in the application:

Listing of Claims

1-39 Canceled

40. (Currently amended) A method for configuring an equivalent $2^n \times 2^n$ k-stage bit-permuting network based on a given $2^n \times 2^n$ k-stage bit-permuting network having ~~the~~ a representation $[\sigma_0 : \sigma_1 : \sigma_2 : \ldots : \sigma_{k-1} : \sigma_k]_n$, the method comprising:

specifying a permutation κ on integers from 1 to n that preserves n, and

implementing the equivalent network as $[\sigma_0 : \sigma_1 : \ldots : \sigma_{j-1}\kappa : \kappa^{-1}\sigma_j : \ldots : \sigma_k]_n$, $j = 1, 2, \ldots,$ or k.

41. (Previously presented) The method as recited in claim 40 wherein the given network is a banyan-type network and the equivalent network is a banyan-type network.

42. (Currently amended) A method for configuring an equivalent $2^n \times 2^n$ k-stage bit-permuting network based on a given $2^n \times 2^n$ k-stage bit-permuting network having ~~the~~ a representation $[\sigma_0 : \sigma_1 : \sigma_2 : \ldots : \sigma_{k-1} : \sigma_k]_n$, the method comprising:

specifying permutations $\kappa_1, \kappa_2, \ldots, \kappa_k$ on integers from 1 to n that preserve n, and

implementing the equivalent network as $[\sigma_0\kappa_1 : \kappa_1^{-1}\sigma_1\kappa_2 : \kappa_2^{-1}\sigma_2\kappa_3 : \ldots : \kappa_{k-1}^{-1}\sigma_{k-1}\kappa_k : \kappa_k^{-1}\sigma_k]_n$.

43. (Previously presented) The method as recited in claim 42 wherein the given network is a banyan-type network and the equivalent network is a banyan-type network.

44. (Currently amended) A method for configuring an equivalent $2^n \times 2^n$ bit-permuting network based on a given $2^n \times 2^n$ bit-permuting network composed of stages and exchanges, the method comprising:

identifying one stage from the stages of the given network, the identified stage having a preceding exchange immediately before it and a succeeding exchange immediately after it,

specifying a permutation on ~~the~~ integers 1 to n that preserves n,

rearranging the preceding exchange and the succeeding exchange with reference to the permutation to generate a rearranged preceding exchange and a rearranged succeeding exchange, respectively, and

implementing the equivalent network so that a stage in the equivalent network corresponding to the identified stage has the rearranged preceding exchange and the rearranged succeeding exchange.

45. (Previously presented) The method as recited in claim 44 wherein the permutation, denoted as κ, induces a $2^n \times 2^n$ cell rearrangement X_κ, and the rearranging includes multiplying the preceding exchange by X_κ from the right-hand side to produce the rearranged preceding exchange and multiplying the succeeding exchange by $X_{\kappa^{-1}}$ from the left-hand side to produce the rearranged succeeding exchange.

46. (Previously presented) The method as recited in claim 45 wherein the given network has k-stages, the given network has the representation $[\sigma_0 : \sigma_1 : \sigma_2 : \ldots : \sigma_{k-1} : \sigma_k]_n$, the identified stage is stage j, and the equivalent network is of the form , $[\sigma_0 : \sigma_1 : \ldots : \sigma_{j-1}\kappa : \kappa^{-1}\sigma_j : \ldots : \sigma_k]_n$, j = 1, 2, …, or k.

47. (Previously presented) The method as recited in claim 44 wherein the given network is a banyan-type network and the equivalent network is a banyan-type network.

48. (Currently amended) A method for configuring an equivalent $2^n \times 2^n$ bit-permuting network by cell rearrangement based on a given $2^n \times 2^n$ bit-permuting network composed of stages and exchanges, the method comprising:

identifying one stage from the stages of the given network, the identified stage having a preceding exchange and a succeeding exchange,

specifying a permutation, denoted as κ, on ~~the~~ integers 1 to n that preserves n and induces a $2^n \times 2^n$ cell rearrangement X_κ,

rearranging the preceding exchange of the identified stage by multiplying the preceding exchange with X_κ from the right-hand side to produce a rearranged preceding exchange and rearranging the succeeding exchange of the identified stage by multiplying the succeeding exchange by $X_{\kappa^{-1}}$ from the left-hand side to produce a rearranged succeeding exchange, and

implementing the equivalent network so that a stage in the equivalent network corresponding to the identified stage has the rearranged preceding exchange and the rearranged succeeding exchange.

49. (Currently amended) A method for cell rearrangement of a $2^n \times 2^n$ bit-permuting network composed of stages and exchanges, the method comprising:

selecting one stage from the stages of the given network to identify a preceding exchange and a succeeding exchange,

specifying a permutation, denoted as κ, on ~~the~~ integers 1 to n that preserves n and induces a $2^n \times 2^n$ cell rearrangement X_κ, and

multiplying the preceding exchange with X_κ from the right-hand side to implement a rearranged preceding exchange and multiplying the succeeding exchange by $X_{\kappa^{-1}}$ from the left-hand side to implement a rearranged succeeding exchange.

50. (Currently amended) A method for cell rearrangement of a given stage of a $2^n \times 2^n$

bit-permuting network composed of stages and exchanges, the method comprising:

specifying a permutation, denoted as κ, on ~~the~~ integers 1 to n that preserves n and induces a $2^n \times 2^n$ cell rearrangement X_κ, and

multiplying ~~the~~ a preceding exchange immediately before the given stage by X_κ from the right-hand side to implement a rearranged preceding exchange for the given stage and multiplying ~~the~~ a succeeding exchange immediately after the given stage exchange by $X_{\kappa^{-1}}$ from the left-hand side to implement a rearranged succeeding exchange for the given stage.

51. (Currently amended) A method for rearranging a given $2^n \times 2^n$ k-stage bit-permuting network having ~~the~~ a representation $[\sigma_0 : \sigma_1 : \sigma_2 : \ldots : \sigma_{k-1} : \sigma_k]_n$ to an equivalent $2^n \times 2^n$ bit-permuting network having the representation $[\pi_0 : \pi_1 : \pi_2 : \ldots : \pi_{k-1} : \pi_k]_n$, the method comprising:

determining permutations $\kappa_1, \kappa_2, \ldots, \kappa_k$ on integers from 1 to n that preserve n, and

implementing the equivalent network with exchanges determined from $\pi_1 = \kappa_1^{-1}\sigma_1\kappa_2$, $\pi_2 = \kappa_2^{-1}\sigma_2\kappa_3$, … , $\pi_{k-1} = \kappa_{k-1}^{-1}\sigma_{k-1}\kappa_k$ so that the equivalent network can be further expressed as $[\alpha : \kappa_1^{-1}\sigma_1\kappa_2 : \kappa_2^{-1}\sigma_2\kappa_3 : \ldots : \kappa_{k-1}^{-1}\sigma_{k-1}\kappa_k : \beta]_n$ for some permutations α and β.

52. (Previously presented) The method as recited in claim 51 wherein the input exchange α of the equivalent network is equal to π_0.

53. (Previously presented) The method as recited in claim 51 wherein the output exchange β of the equivalent network is equal to π_k.

54. (Previously presented) The method as recited in claim 51 wherein the input exchange α of the equivalent network is equal to π_0 and the output exchange β of the equivalent network is equal to π_k.

55. (Currently amended) A method for configuring a given $2^n \times 2^n$ k-stage bit-permuting

network to achieve a desired trace, the method comprising:

determining a permutation σ on ~~the~~ integers 1 to n that maps ~~the~~ a trace of the given network term-by-term to the desired trace, and

prepending the given network with an extra input exchange induced by σ^{-1} if the permutation σ exists.

56. (Previously presented) A method as recited in claim 55 wherein k = n and the bit-permuting network is a $2^n \times 2^n$ banyan-type network.

57. (Previously presented) A method as recited in claim 55 wherein the trace of the given network is the sequence $t_1, t_2, \ldots, t_k$, the desired trace is the sequence $t'_1, t'_2, \ldots, t'_k$, and $t'_j = \sigma(t_j)$ for $j = 1, 2, \ldots, k$.

58. (Currently amended) A method for configuring a given $2^n \times 2^n$ k-stage bit-permuting network to achieve a desired guide, the method comprising:

determining a permutation π on ~~the~~ integers 1 to n that maps ~~the~~ a guide of the given network term-by-term to the desired guide, and

appending the given network with an extra output exchange induced by π if the permutation π exists.

59. (Previously presented) A method as recited in claim 58 wherein k = n and the bit-permuting network is a $2^n \times 2^n$ banyan-type network.

60. (Previously presented) A method as recited in claim 58 wherein the guide of the given network is the sequence $g_1, g_2, \ldots, g_k$, the desired guide is the sequence $g'_1, g'_2, \ldots, g'_k$, and $g'_j = \pi(g_j)$ for $j = 1, 2, \ldots, k$.

61. (Currently amended) A method for configuring a given $2^n \times 2^n$ k-stage bit-permuting network to achieve a desired trace and a desired guide, the method comprising:

determining a permutation σ on ~~the~~ integers 1 to n that maps ~~the~~ a trace of the given network term-by-term to the desired trace,

determining a permutation π on the integers 1 to n that maps ~~the~~ a guide of the given network term-by-term to the desired guide, and

if both the permutations σ and π exist, prepending the given network with an extra input exchange induced by σ^{-1}, and appending the given network with an extra output exchange induced by π.

62. (Previously presented) A method as recited in claim 61 wherein k = n and the bit-permuting network is a $2^n \times 2^n$ banyan-type network.

63. (Previously presented) A method as recited in claim 61 wherein the trace of the given network is the sequence $t_1, t_2, \ldots, t_k$, the desired trace is the sequence $t'_1, t'_2, \ldots, t'_k$, and $t'_j = \sigma(t_j)$ for $j = 1, 2, \ldots, k$ and wherein the guide of the given network is the sequence $g_1, g_2, \ldots, g_k$, the desired guide is the sequence $g'_1, g'_2, \ldots, g'_k$, and $g'_j = \pi(g_j)$ for $j = 1, 2, \ldots, k$.

64. (Currently amended) A method for rearranging a given $2^n \times 2^n$ banyan-type network having ~~the~~ a representation $[\sigma_0 : \sigma_1 : \sigma_2 : \ldots : \sigma_{n-1} : \sigma_n]_n$ to an equivalent $2^n \times 2^n$ banyan-type network having the representation $[\pi_0 : \pi_1 : \pi_2 : \ldots : \pi_{n-1} : \pi_n]_n$, the method comprising:

determining permutations $\kappa_1, \kappa_2, \ldots, \kappa_n$ on integers from 1 to n that preserve n, and

implementing the equivalent network with exchanges determined from $\pi_1 = \kappa_1^{-1}\sigma_1\kappa_2$, $\pi_2 = \kappa_2^{-1}\sigma_2\kappa_3$,, $\ldots$, $\pi_{n-1} = \kappa_{n-1}^{-1}\sigma_{n-1}\kappa_n$ so that the equivalent network can be further expressed as $[\alpha : \kappa_1^{-1}\sigma_1\kappa_2 : \kappa_2^{-1}\sigma_2\kappa_3 : \ldots : \kappa_{n-1}^{-1}\sigma_{n-1}\kappa_n : \beta]_n$ for some permutations α and β.

65. (Previously presented) The method as recited in claim 64 wherein the input

exchange α of the equivalent network is equal to π_0.

66. (Previously presented) The method as recited in claim 64 wherein the output exchange β of the equivalent network is equal to π_k.

67. (Previously presented) The method as recited in claim 64 wherein the input exchange α of the equivalent network is equal to π_0 and the output exchange β of the equivalent network is equal to π_k.

68. (Currently amended) A method for rearranging a first $2^n \times 2^n$ banyan-type network having ~~the~~ a representation $[\sigma_0 : \sigma_1 : \ldots : \sigma_{n-1} : \sigma_n]$ with a first trace induced by a permutation τ on integers 1 to n and a first guide induced by a permutation γ on integers 1 to n to a second $2^n \times 2^n$ banyan-type network having the representation $[\lambda\sigma_0 : \sigma_1 : \ldots : \sigma_{n-1} : \sigma_n\pi]$, the method comprising:

prepending an additional input exchange X_λ to the first network, and

appending an additional output exchange X_π to the first network, wherein the second network is characterized by a second trace induced by a permutation τ' on integers 1 to n and a second guide induced by a permutation γ' on integers 1 to n such that $\tau' = \tau\lambda^{-1}$ and $\gamma' = \gamma\pi$.

69. (Previously presented) The method as recited in claim 68 wherein the permutations τ and γ that induce the first trace and the first guide are converted to any τ' and γ', respectively, with the prepended input exchange X_λ and the appended output exchange X_π by computing $\lambda = \tau'^{-1}\tau$ and $\pi = \gamma^{-1}\gamma'$.

70. (Currently amended) A method for configuring a given $2^n \times 2^n$ banyan-type network to achieve a desired trace wherein ~~the~~ a trace of the given network is induced by a permutation τ on integers 1 to n, and the desired trace is induced by another permutation τ' on integers 1 to n,

the method comprising:

determining a permutation $\lambda = \tau'^{-1}\tau$, and

prepending the given network with an extra input exchange induced by λ.

71. (Previously presented) A method as recited in claim 70 wherein the desired trace is 1, 2, …, n and the permutation $\lambda = \tau$.

72. (Previously presented) A method as recited in claim 70 wherein the desired trace is n, n−1, …, 1 and the permutation $\lambda = \sigma_{\leftrightarrow}^{(n)}\tau$.

73. (Currently amended) A method for configuring a given $2^n \times 2^n$ banyan-type network to achieve a desired guide wherein ~~the~~ a guide of the given network is induced by a permutation γ on integers 1 to n, and the desired guide is induced by another permutation γ' on integers 1 to n, the method comprising:

determining a permutation $\pi = \gamma^{-1}\gamma'$, and

appending the given network with an extra output exchange induced by π.

74. (Previously presented) A method as recited in claim 73 wherein the desired guide is 1, 2, …, n and the permutation $\pi = \gamma^{-1}$.

75. (Previously presented) A method as recited in claim 73 wherein the desired guide is n, n−1, …, 1 and the permutation $\pi = \gamma^{-1}\sigma_{\leftrightarrow}^{(n)}$.

76. (Currently amended) A method for configuring a given $2^n \times 2^n$ banyan-type network to achieve a desired trace and a desired guide wherein ~~the~~ a trace of the given network is induced by a permutation τ on integers 1 to n, the desired trace is induced by another permutation τ' on

integers 1 to n, ~~the~~ a guide of the given network is induced by a permutation γ on integers 1 to n, and the desired guide is induced by another permutation γ' on integers 1 to n, the method comprising:

determining a permutation $\lambda = \tau'^{-1}\tau$,

determining a permutation $\pi = \gamma^{-1}\gamma'$,

prepending the given network with an extra input exchange induced by λ, and

appending the given network with an extra output exchange induced by π.

77. (Currently amended) An equivalent $2^n \times 2^n$ k-stage bit-permuting network based on a given $2^n \times 2^n$ k-stage bit-permuting network having ~~the~~ a representation $[\sigma_0 : \sigma_1 : \sigma_2 : \ldots : \sigma_{k-1} : \sigma_k]_n$ the equivalent network comprising:

permutation means for computing a permutation κ on integers from 1 to n that preserves n, and

a $2^n \times 2^n$ k-stage bit-permuting network configured as $[\sigma_0 : \sigma_1 : \ldots : \sigma_{j-1}\kappa : \kappa^{-1}\sigma_j : \ldots : \sigma_k]_n$, $j = 1, 2, \ldots$, or k.

78. (Currently amended) An equivalent $2^n \times 2^n$ k-stage bit-permuting network based on the j-th stage of a given $2^n \times 2^n$ k-stage bit-permuting network having ~~the~~ a representation $[\sigma_0 : \sigma_1 : \sigma_2 : \ldots : \sigma_{k-1} : \sigma_k]_n$ and based on a permutation κ on integers from 1 to n that preserves n, the equivalent network comprising:

an input exchange $\sigma_0\kappa$ if j=l, or an input exchange σ_0 if $j = 2, 3, \ldots, k$,

an output exchange $\kappa^{-1}\sigma_k$ if j=k, or an output exchange σ_k if $j=1, 2, \ldots, k-1$, and

interstage exchanges $\sigma_1, \sigma_2, \ldots, \sigma_{j-1}\kappa, \kappa^{-1}\sigma_j, .., \sigma_{k-1}$ if $j = 2, \ldots$, or $k - 1$, or interstage exchanges $\kappa^{-1}\sigma_1, \sigma_2, \ldots, \sigma_j, \ldots, \sigma_{k-1}$ if $j = 1$, or interstage exchanges $\sigma_1, \sigma_2, \ldots, \sigma_j, .., \sigma_{k-2}, \sigma_{k-1}\kappa$ if $j = k$.